Typing Speed Test este un joc util daca vrei sa iti testezi ablilitatea de a tasta repede, dar si un mod bun de a exersa.

Pentru implementarea jocului am folosit libraria pygame si am importat si module precum sys, time si random.

Clasa Game contine toate functile necesare pornirii si resetarii jocului si, implicit, si functiile necesarii bunei functionari a acestuia.

In constructor am definit toate variabilele care ne-au fost de folos in crearea jocului, variabila “screen” fiind cea mai importanta dintre ele. Am setat latimea si inaltimea ferestrei la 750, respectiv 500, am initializat pygame si am incarcat imaginile de fundal. Jocul va porni automat dupa un countdown de 3 secunde.

Metoda draw\_text() este cea cu ajutorul careia am desenat textul pe ecran. Argumentele pe care aceasta le primeste sunt “screen”, “msg”- mesajul pe care vrem sa il afisam pe ecran, coordonata y la care dorim sa fie pozitionat, dimensiunea fontului, “fsize” si culoarea acestuia “color”. Dupa scrierea textului dorit in centrul ferestrei, am apelat metoda update() pentru a actualiza ecranul conform modificarilor facute.  
 Metoda draw\_text\_icon() este utilizata pentru pozitionarea corecta a butoanelor de “Retry”, respectiv “Next level”. Spre deosebire de metoda prezentata anterior, aceasta permite si pozitionarea in functie de coordonata x.

Metoda get\_sentence() deschide fisierul din care sunt preluate propozitiile in functie de butonul apasat de jucator, astfel:

Prin apasarea butonului “Retry”, textul este preluat random din fisierul “sentences.txt” care contine propozitii pentru nivelul “incepator”.

Prin apasarea butonului “Next level”, textul afisat pe ecran este preluat random din fisierele “b”si “c”, care contin propozitii mai lungi si mai complicate, corespunzatoare nivelelor “avansat”, respectiv “experimentat”.

Metoda show\_results() este cea in care am calculat viteza de tastare a jucatorului. Preluam timpul la care jucatorul a apasat click pe casuta de input si timpul la care a apasat tasta “Enter”, facem diferenta si obtinem rezultatul in secunde.

Pentru calculul acuratetei am comparat caracterele introduse in casuta de input si le-am comparat cu cele din propozitia afisata pe ecran, numarand caracterele corecte si stocandu-le intr-o variabila count. In final, pentru a returna acuratetea, am utilizat formula (caractere corecte \* 100)/nr total de caractere din propozitie.

WPM – cuvintele pe minut. Am considerat ca majoritatea cuvintelor sunt formate din aproximativ 5 caractere si am calculat numarul de cuvinte pe minut prin impartirea numarului total de cuvinte la 5 si prin impartirea acestui rezultat la timpul total scris in minute. Deoarece initial am calculat timpul in secunde, aici a fost nevoie sa facem o conversie prin impartirea acestuia la 60.

Tot in aceasta metoda desenam si butoanele de Retry si Next level care apar pe ecran dupa ce jucatorul a apasat tasta Enter. Apasand butonul de Retry jocul va fi reluat, iar apasand butonul de Next level jucatorul va trece la nivelul urmator care are un nivel de dificultate mai ridicat.

Metoda run() este metoda principala care face posibila realizarea tuturor evenimentelor. La inceputul acesteia este apelata metoda reset\_game() care reseteaza toate valorile variabilelor. In continuare am folosit o bucla while in care sunt capturate evenimentele aparute la apasarea tastelor sau a click-ului. In interiorul while-ului am folosit un for in care urmarim evenimentele care au fost produse de apasarea mouse-ului sau a unei taste. Verificam pozitia cursorului de fiecare data cand jucatorul apasa click, daca acesta este pe casuta de input pornim timpul si setam variabile active ca fiind True, care permite introducerea caracterelor de catre utilizator. Daca acesta se afla pe butonul “Retry”, jocul este resetat, iar daca se afla pe butonul “Next level” jucatorul va trece la nivelul urmator, in care nivelul de dificultate al propozitiilor afisate va fi mai mare.

Cat timp variabila active este True iar jucatorul nu s-a oprit din tastat urmarim evenimentele generate de apasarea tastelor. La orice apasare de tasta, casuta de input va fi actualizata. Tasta “Enter” semnifica terminarea introducerii textului si odata apasata punctajul rundei va fi calculat si afisat pe ecran. Un alt eveniment este generat de apasarea tastei BackSpace care sterge ultimul caracter inserat in casuta de text.

Metoda reset\_game() reseteaza toate variabilele pentru ca jucatorul sa isi poata testa din nou abilitatile. In aceasta metoda este apelata metoda get\_sentence() care selecteaza random o propozitie din fisierul corespunzator nivelului.

In final, am creat obiectul Game() pentru a putea rula programul.